

# SUPPLEMENT

## TO BRITISH TELECOMMUNICATIONS ENGINEERING

(formerly the Supplement to The Post Office Electrical Engineers' Journal)

Vol. 6 Part 2 July 1987

### Contents

ISSN 0262-4028

### BTEC & CGLI GUIDANCE FOR STUDENTS

BTEC: MICROELECTRONIC SYSTEMS II .. . . .	17
CGLI: TRANSMISSION T4 OPTION (1986) . . . . .	23
CGLI: MICROELECTRONIC SYSTEMS T4 OPTION (1986)...	28

## BUSINESS AND TECHNICIAN EDUCATION COUNCIL

### National Certificate in Telecommunications

Sets of model questions and answers for Business and Technician Education Council (BTEC) units are given below. The questions illustrate the types of questions that students may encounter, and are useful as practice material for the skills learned during the course.

Where additional text is given for educational purposes, it is shown within square brackets to distinguish it from information expected of students under examination conditions. Representative time limits for questions are shown, and care has been taken to give model answers that reflect these limits.

We would like to emphasise that the questions are not representative of questions set by any particular college.

#### BTEC: MICROELECTRONIC SYSTEMS II

The following questions are based on the BTEC's standard unit U79/603. Students are advised to read the notes above.

#### SECTION A

Questions 1–12 are multiple-choice questions. For each question, students are required to indicate which of the four alternatives, (a), (b), (c) or (d), correctly answers the question or correctly completes the unfinished statement given. Approximately 1½ minutes should be allowed for each of these questions. Where appropriate, the questions refer to 8 bit microprocessors.

**Q1** Most digital micro-electronic systems use two-state logic elements. This means that ...

(a) most digital computers can hold and manipulate numbers coded only in the binary number system.

(b) most digital computers can hold and manipulate any pattern of binary states.

(c) machine-code instructions can be expressed only as binary numbers.

(d) all non-numeric data must be expressed in the form of binary numbers.

**A1** (b) most digital computers can hold and manipulate any pattern of binary states.

[Tutorial Note: Any system of codes or numbers can be used depending on how the machine has been programmed.]

**Q2** Numbers in two-state logic systems:

(a) can be expressed only by using the binary number system.

(b) can be only positive numbers.

(c) may take any value depending upon the code and the number of digits.

(d) cannot have a radix or base other than two.

**A2** (c) may take any value depending upon the code and the number of digits.

[Tutorial Note: Any number system can be coded in 1s and 0s; for example, binary coded decimal (BCD).]

**Q3** Denary number systems are not particularly suitable for direct use in digital computer systems because ...

(a) digital computer systems usually use two-state signals.

(b) digital electronic systems can have only two states.

(c) digital circuits cannot deal with multilevel signals.

(d) the denary number system cannot be adapted for use in binary systems.

**A3** (a) digital computer systems usually use two-state signals.

**Q4** The hexadecimal system has a base of ...

(a) 8

(b) 10

(c) 12

(d) 16

**A4** (d) 16.

**Q5** A group of bistables or flip-flop circuits, used for the temporary storage of data within a microprocessor, is known collectively as ...

(a) a logic gate.

(b) a read-only memory.

(c) a register.

(d) a data bus.

**A5** (c) a register.

**Q6** A single binary byte can represent denary integers in the continuous and inclusive range ...

(a) 0 to +256.

(b) -256 to +256.

(c) -128 to +128.

(d) 0 to +255.

**A6** (d) 0 to +255.

[Tutorial Note: If signed binary is used, the range would be 1000 0000 to 0111 1111, which is -128 to +127.]

**Q7** An individual bistable or flip-flop circuit, used by a microprocessor to indicate a particular internal condition, is known as ...

(a) an accumulator.

(b) a pointer.

(c) a status register.

(d) a status flag.

A7 (d) a status flag.

[Tutorial Note: The term *status register* refers to a group of status flags.]

Q8 The operand of an instruction is usually that part of the instruction ...

- (a) fetched into the instruction register.
- (b) which specifies the internal register to be used.
- (c) which specifies the type of operation to be carried out.
- (d) which specifies the data or location to be used in the operation.

A8 (d) which specifies the data or location to be used in the operation.

[Tutorial Note: An operation code or 'op-code' consists of an operator and generally an operand. The operator can be regarded as 'what to do' and the operand as 'what with or where'. The register to be used and the type of operation is usually implicit in the operator part of the instruction.]

Q9 A program counter (PC) in an 8 bit microprocessor such as the Zilog Z80 is ...

- (a) an 8 bit register containing the memory address of the next step in the program.
- (b) a 16 bit register containing the memory address of the next step in the program.
- (c) a register containing the current byte of data, required by the program.
- (d) a register which stores additions performed by the program.

A9 (b) a 16 bit register containing the memory address of the next step in the program.

Q10 The results of arithmetic operations are stored within a microprocessor in the ...

- (a) program counter.
- (b) data bus.
- (c) accumulator.
- (d) random-access memory.

A10 (c) accumulator.

Q11 The section of a microprocessor where operations such as binary addition takes place is called the ...

- (a) instruction decoder.
- (b) ALU.
- (c) accumulator.
- (d) instruction register.

A11 (b) ALU.

[Tutorial Note: ALU stands for arithmetic and logic unit.]

Q12 After each instruction fetch, the next operation performed by a microprocessor is ...

- (a) to decode the instruction.
- (b) to increment the program counter.
- (c) to execute the instruction.
- (d) to fetch the data.

A12 (b) to increment the program counter.

[Tutorial Note: The program counter is automatically incremented immediately after each fetch.]

## SECTION B

Questions 13–21 are short-answer questions. Approximately six minutes should be allowed for each of these questions.

Q13 State THREE advantages of the octal system and THREE advantages of the hexadecimal system for digital computer applications.

## A13 Octal

Radix is a power of two and therefore numbers in octal relate more easily to binary than those in denary.

Numbers and codes in octal are easier for human operators to read and recognise than those in binary.

Octal requires fewer digits than binary for a given range of codes or numbers, and is therefore more compact.

## Hexadecimal

Radix is a power of two and therefore numbers in hexadecimal relate more easily to binary than those in denary.

Numbers and codes in hexadecimal are easier for human operators to read and recognise than are those in binary.

Hexadecimal requires fewer digits than binary, octal or denary to represent a given range of codes or numbers and all 8 bit words can be represented by two hexadecimal digits.

Q14 Complete the following table summarising the rules of binary addition.

A	+	B	+	Carry In	=	Sum	+	Carry Out
0		0		0				
0		0		1				
0		1		0				
0		1		1				
1		0		0				
1		0		1				
1		1		0				
1		1		1				

## A14

A	+	B	+	Carry In	=	Sum	+	Carry Out
0		0		0		0		0
0		0		1		1		0
0		1		0		1		0
0		1		1		0		1
1		0		0		1		0
1		0		1		0		1
1		1		0		0		1
1		1		1		1		1

Q15 Complete the following table showing the rules of binary subtraction.

A	–	B	–	Borrow In	=	Difference	–	Borrow Out
0		0		0				
0		0		1				
0		1		0				
0		1		1				
1		0		0				
1		0		1				
1		1		0				
1		1		1				

## A15

A	–	B	–	Borrow In	=	Difference	–	Borrow Out
0		0		0		0		0
0		0		1		1		1
0		1		0		1		1
0		1		1		0		1
1		0		0		1		0
1		0		1		0		0
1		1		0		0		0
1		1		1		1		1

**Q16** Convert the following numbers to their 8 bit unsigned binary values.

- (a) 205 base 10,  
(b) 232 base 8, and  
(c) 6B base 16.

**A16** (a) [Tutorial Note: A denary number can be converted to binary by successively dividing by 2 and writing down the remainders.]

	Remainder
2)205	1
2)102	0
2) 51	1
2) 25	1
2) 12	0
2) 6	0
2) 3	1
2) 1	1
0	

Thus 205 denary is 1100 1101 in 8 bit unsigned binary.

(b) [Tutorial Note: An octal number can be converted to binary by simply writing down the binary equivalents of each digit.]

2	3	2
↓	↓	↓
010	011	010

Thus 232 octal is 1001 1010 in 8 bit unsigned binary.

(c) [Tutorial Note: A hexadecimal number can be converted in a similar manner.]

6	B
↓	↓
0110	1011

Thus 6B hexadecimal is 0110 1011 in 8 bit unsigned binary.

**Q17** Convert the following numbers to their 8 bit signed binary, two's complement form.

- (a) -100 base 10,  
(b) -67 base 8, and  
(c) -5C base 16.

**A17** [Tutorial Note: The answers are derived by firstly converting the numbers into 8 bit binary form (adding leading zeros as necessary), then converting to one's complement form by inverting each digit, and then adding 1 to obtain the two's complement.]

(a)

	Remainder
2)100	0
2) 50	0
2) 25	1
2) 12	0
2) 6	0
2) 3	1
2) 1	1
0	

Thus in 8 bit binary 100 denary is 0110 0100.  
Inverting each bit and adding 1:

0110 0100	
1001 1011	Invert
1	Add 1
1001 1100	

Thus the result is 1001 1100.

(b) 67 octal is 0011 0111 in 8 bit unsigned binary.

0011 0111	
1100 1000	Invert
1	Add 1
1100 1001	

Thus the result is 1100 1001.

(c) 5C hexadecimal is 0101 1100 in 8 bit unsigned binary.

0101 1100	
1010 0011	Invert
1	Add 1
1010 0100	

Thus the result is 1010 0100.

**Q18** Convert the following numbers to their equivalent form in the base stated.

- (a) 75 denary to octal,  
(b) 3.75 denary to binary,  
(c) 101 denary to hexadecimal,  
(d) 8F hexadecimal to denary,  
(e) 3C hexadecimal to octal,  
(f) 37 octal to denary, and  
(g) 74 octal to hexadecimal.

**A18** (a)

	Remainder
8)75 <sub>10</sub>	3
8) 9 <sub>10</sub>	1
8) 1	1
0	

Thus, 75 denary = 113 octal.

(b) [Tutorial Note: The binary equivalent of the integral part is 11. The binary equivalent of the fractional part is found by multiplying the fractional part by 2, writing down the integral part of the result and then multiplying the fractional part by 2, and so on until the resulting fractional part is 0.]

Integral Part	Fractional Part
	$.75 \times 2$
1	$.50 \times 2$
1	$.00$

Thus, 3.75 denary = 11.11 binary.

(c) 101 denary =  $6 \times 16 + 5 =$  65 hexadecimal.

(d) 8F hexadecimal =  $8 \times 16 + 15 =$  143 denary.

(e) 3C hexadecimal =  $3 \times 16 + 12 =$  60 denary = 74 octal.

(f) 37 octal =  $3 \times 8 + 7 =$  31 denary.

(g) 74 octal =  $7 \times 8 + 4 =$  60 denary = 3C hexadecimal.

**Q19** Show how 37 denary can be subtracted from 52 denary in binary by complementary addition using 8 bit two's-complement signed binary.

**A19** 52 denary = 0011 0100 binary

37 denary = 0010 0101 binary

One's complement of 37 denary = 1101 1010 binary

Add 1 + 1

Two's complement of 37 denary = 1101 1011 binary

Add 0011 0100 (+52)

and 1101 1011 (-37)

gives 0000 1111 binary

which is 15 denary.

**Q20** Give the results of the following additions and subtractions:

- (a) 0010 1101 + 0110 0011 in binary.  
(b) 0100 1100 - 0011 0011 in binary.  
(c) 3E + 2A in hexadecimal.  
(d) A4 - 3D in hexadecimal.

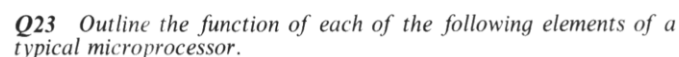
- Q21** State the type of addressing mode used in the following operations. (LD can be taken as meaning LOAD, STORE or MOVE, accordingly.)

- A21** (a) Direct  
(b) Immediate  
(c) Indirect  
(d) Indexed.

Questions 22–27 are longer questions typical of a final assessment. Approximately 20 to 30 minutes should be allowed for each of these questions.

- the arithmetic and logic unit (ALU) and its associated registers;
- the instruction register, decoder, and control unit;
- the general- and special-purpose registers; and
- the data, address and control buses and buffers.

A22



- (a) the arithmetic and logic unit (ALU),
- (b) the accumulator,
- (c) the store address register or registers,
- (d) the program counter, and
- (e) the stack pointer.

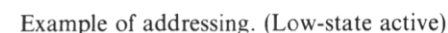
20

[*Tutorial Note:* A universal shift register is a register whose contents can be shifted from either left to right or from right to left, and data can be read into it or out of it either serially or in parallel form; this allows a wide range of manipulations.]

(d) The stack pointer (SP) is a memory address register used for the specific purpose of holding the current address of the location of the 'top' of the stack. The stack is a consecutive part of memory that is used for the temporary storage of addresses and data. It uses the principle of a 'last in-first out' list in which items requiring temporary storage are 'pushed' onto the top of the list or 'popped' off the top of the list. The SP holds the address of the last item on the stack (or in some cases the address of the next available address) and is automatically incremented or decremented as items are 'pushed' and 'popped'.

- In particular show how each memory element is addressed by means of binary address decoders. Give an example of how a particular memory element or cell can be addressed.

- A24** See sketch.



Address		Row	Column
A3   A2	A1   A0	0   1   2   3	0   1   2   3
1   0	1   1	1   1   0   1	1   1   1   0
row	column		

↑

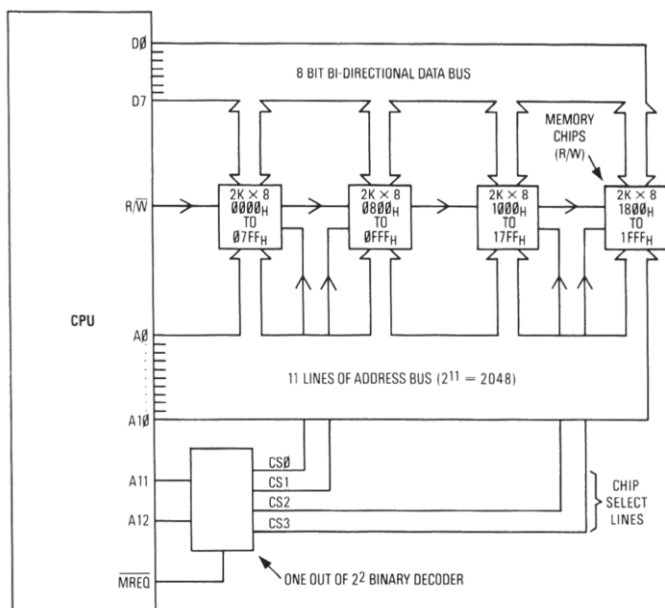
row 2  
selected

↑

column 3  
selected

**Q25** A microcomputer system has FOUR memory integrated circuits (ICs) each of  $2048 \times 8$  bit. Sketch a diagram showing how the four ICs would be connected to the system address and data buses and how chip selection would be achieved.

**A25**



(b) *Change in the Number of Lines* A peripheral such as a printer can be operated by means of serial data, whereas the data available on the data bus of the computer system is usually in parallel form. Thus an interface device may be needed to convert between serial and parallel data or vice versa.

(c) *Timing Control* Peripherals can rarely send or receive data at the speed at which a typical microprocessor can handle data. It is therefore necessary for the interface unit to control the rate at which data is sent or received by means of 'handshaking' signals; that is, control signals that indicate when the peripheral is ready to send or receive the next item of data, etc.

(d) *Port Specification and Data Direction* A typical microcomputer system has a single bidirectional data bus and thus, if data is to be sent or received to or from a number of devices or peripherals, it may be necessary for the interface unit to select the input/output port or line that is required at any instant. This can involve 'addressing' the particular port required and setting data direction registers within the interfacing device, not only to choose the lines required, but to designate them for the input of data or the output of data.

[Tutorial Note: Other functions might include code conversion, data buffering, multiplexing, modulation and demodulation, etc.]

## SECTION D

Questions 28-30 inclusive are concerned with microprocessor programming in machine code and may use the instructions given in Table 1.

[Tutorial Note: This table is a selection taken from the Z80 instruction set. Students should expect that a similar selection could be taken from the instruction set of other types of microprocessor.]

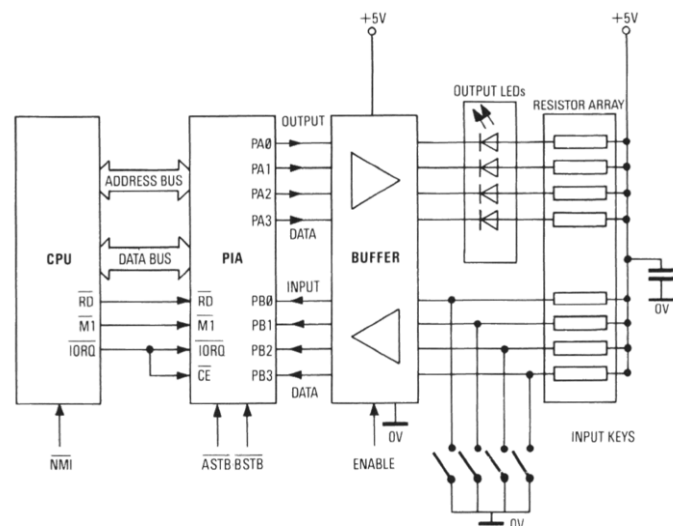
**Table 1**

Instruction	Hexadecimal Code	Mnemonic Code
Load A-reg with byte <i>n</i>	3E	LD A, <i>n</i>
Load B-reg with byte <i>n</i>	06	LD B, <i>n</i>
Load HL-reg with 2 bytes <i>nn</i>	21	LD HL, <i>nn</i>
Load DE-reg with 2 bytes <i>nn</i>	11	LD DE, <i>nn</i>
Load A-reg with the byte from address (HL)	7E	LD A, (HL)
Load A-reg with the byte from address ( <i>nn</i> )	3A	LD A, ( <i>nn</i> )
Load B-reg with the byte from address (HL)	46	LD B, (HL)
Load address (HL) from the A-reg	77	LD (HL), A
Load address (DE) from A-reg	12	LD (DE), A
Load address (HL) from the B-reg	70	LD (HL), B
Load address ( <i>nn</i> ) from the A-reg	32	LD ( <i>nn</i> ), A
Add B-reg to A-reg	80	ADD A, B
Subtract B-reg from A-reg without carry	90	SUB B
Compare B-reg with A-reg and set the flag registers	B8	CP B
Increment HL-reg	23	INC HL
Decrement HL-reg	2B	DEC HL
Increment DE-reg	13	INC DE
Jump to address <i>nn</i>	C3	JP <i>nn</i>
Jump to address <i>nn</i> if zero flag is set	CA	JPZ <i>nn</i>
Jump to address <i>nn</i> if zero flag is not set	C2	JPNZ <i>nn</i>
Decrement B-reg	05	DEC B
Decrement B-reg and jump relative by <i>n</i> if not zero	10	DJNZ <i>n</i>
Halt	76	HALT

reg: register

**Q26** Sketch a block diagram showing how a microprocessor system consisting of a control processing unit (CPU) and a peripheral interface adaptor (PIA) can be connected to four input keys and four light-emitting diode (LED) output indicators via data buffers.

**A26**

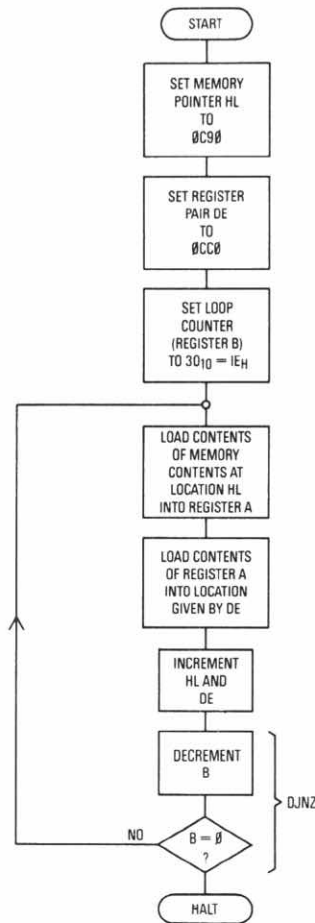


[Note: The control line and handshaking connections would differ depending on the devices used and the arrangements required.]

**Q27** Give examples of FOUR typical functions of interfacing devices that can be used to connect a peripheral to a microcomputer system. Briefly explain the function in each case.

**A27** (a) *Electrical Buffering* The current and/or voltage required for operation of the peripheral may be different from that available from the internal buses of the microcomputer system; for example, 20 mA current loop drivers may be needed.

**Q28** A block of 30 (denary) numbers is stored in memory starting at address 0C90 hex. Fig. 1 shows a flow chart of a program to move the block of numbers to a new starting address of 0CC0 hex. By using the instruction set provided in Table 1, write a program in source and object code starting at address 0C00 to perform the block move.



A31

Address	Object code	Label	Source Code	Comments
0C00	21 90 0C	Start:	LD HL, 0C90H	; set pointer
0C03	11 C0 0C		LD DE, 0CC0H	; set second ; pointer
0C06	06 1E	Loop:	LD B, 1EH	; set counter
0C08	7E		LD A, (HL)	; load data
0C09	12		LD (DE), A	; store data
0C0A	23		INC HL	
0C0B	13		INC DE	
0C0C	10 FA		DJNZ FA	; Dec B and jump ; relative by ; minus ; 6 if B ≠ 0
0C0E	76	Halt		

[Tutorial Note: The relative jump at 0C0C (that is, DJNZ FA) is a jump from PC setting 0C0E to 0C08 which is minus 6. In hexadecimal, this is FA since  $FA + 6 = 00$ . It should be remembered that the PC has been automatically incremented after fetching the DJNZ F8 instruction.]

**Q29** The following is a segment of a machine-code program. Outline the separate sequential states of operation involved in fetching and executing the instruction contained in the segment.

Address	Object Code	Source Code
0100	3A	LD A, (0200H)
0101	00	
0102	02	
0200	05	; data

Program counter (PC) = 0100.

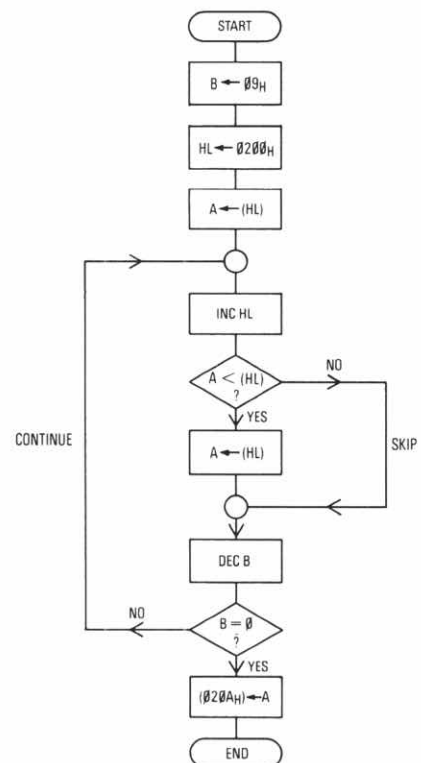
A29

- 1 PC contents (0100) to address bus.
- 2 Instruction (3A) fetched from (0100) on data bus to instruction register (IR).
- 3 PC incremented to 0101.
- 4 Instruction (3A) decoded—three more fetches are required.
- 5 PC contents (0101) to address bus.
- 6 Low byte of operand (00) fetched from address 0101 on data bus to temporary address register.
- 7 PC incremented to 0102.
- 8 PC contents (0102) to address bus.
- 9 High byte of operand (02) fetched from address 0102 on data bus to temporary address register.
- 10 PC incremented to 0103.
- 11 Address (0200) from temporary address register to address bus.
- 12 Data (05) fetched from address 0200 on data bus to accumulator.

**Q30** Write an algorithm and draw a flow chart suitable for a machine-code program to find the largest value of a set of 10 data items stored at consecutive memory addresses from 0200H to 0209H inclusive. The result is to be stored at address 020AH.

- A30**
- 1 Set B-register to 09 (counter).
  - 2 Set HL-register to 0200H (data pointer).
  - 3 Load A-register from (HL), i.e. first address 0200H.
  - 4 Increment HL; point to next data item.
  - 5 Compare (HL) with A-register; compare next data item with the previous.
  - 6 If A is larger than (HL), then skip to step 8.
  - 7 Load A-register from (HL); keep the larger.
  - 8 Decrement B-register (counter).
  - 9 If B is not zero, then jump to step 4.
  - 10 Store contents of A-register (largest) at address 020AH.
  - 11 End.

The flow chart is given in the sketch.



Questions and answers contributed by B. Evans

# CITY AND GUILDS OF LONDON INSTITUTE

## Telecommunications Technicians (New) Scheme

The following questions are from examination papers set for the City and Guilds of London Institute's (CGLI's) new 271 Telecommunications Technicians Scheme, and are reproduced with the permission of the CGLI. The answers given have been prepared by independent authors. Answers to some questions are occasionally omitted because of insufficient space. Students studying BTEC courses may find that these questions are useful for revision.

### CGLI: TRANSMISSION T4 OPTION (1986)

Students were required to answer any six questions. The time allowed was three hours. Students are advised to read the notes above

**Q1** For a multi-pair cable

(a) Explain, with the aid of a sketch, what is meant by

- (i) near-end crosstalk
- (ii) far-end crosstalk.

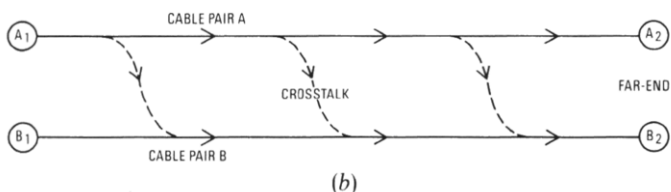
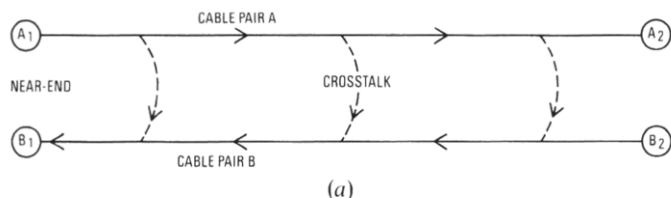
(b) State *FOUR* sources of crosstalk.

(c) Explain how crosstalk may be minimised during

- (i) manufacture
- (ii) installation.

**A1** (a) Crosstalk in multi-pair cables is unwanted coupling of signals between pairs.

(i) Sketch (a) shows near-end crosstalk where a signal propagating from  $A_1$  to  $A_2$  is coupled into cable pair B; that part of the coupled signal which travels back towards  $B_1$  is termed *near-end crosstalk*. The resultant unwanted signal at  $B_1$  consists of components which have travelled different distances, the signal is therefore a 'distorted' version of the signal propagating in cable pair A.



(ii) Sketch (b) shows far-end crosstalk. The crosstalk is due to that part of the coupled signal which travels in the forward direction to  $B_2$ . All components of the coupled signal cover the same distance, and the resultant unwanted signal at  $B_2$  is an 'undistorted' version of the signal propagating in A.

(b) Four possible sources of crosstalk are:

- (i) capacitive coupling,
- (ii) inductive coupling,
- (iii) coupling due to insulation leakage, and
- (iv) unbalanced circuits.

(c) Crosstalk can be minimised by careful design. Use of dielectric materials with high insulation resistance and low permittivity keep leakage and capacitive coupling to a minimum. Maximising uniformity keeps any unbalance to a minimum. Here the use of conductors made from the same die and adopting cable identification techniques which ensure that the same amount of marking ink is used per unit length help maintain uniformity. Interference between quads can be reduced by using different lay lengths and by alternating twist directions.

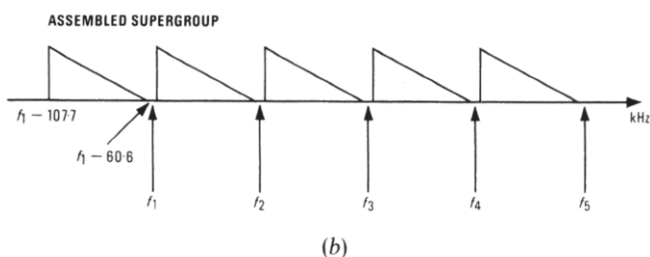
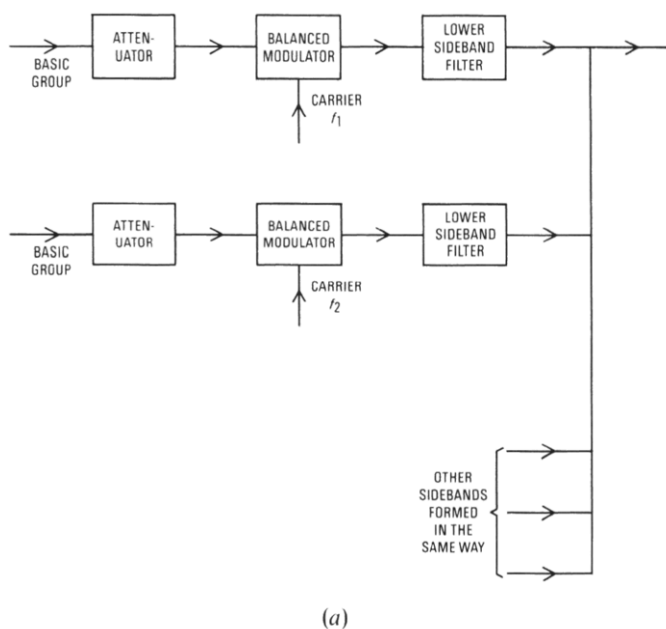
(ii) During installation, cable joints and terminations should be tested and any unbalance corrected. All jointing should be protected against the ingress of water or damp air into the cable system.

**Q2** (a) Explain with the aid of a block diagram how *FIVE* basic groups are group-modulated to form a supergroup.

(b) State the carrier frequencies used in (a).

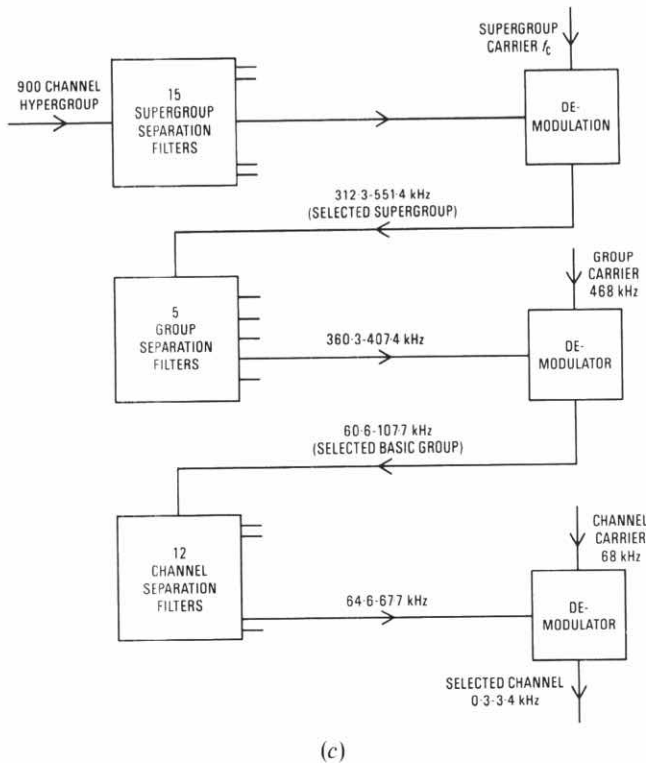
(c) Draw and fully label a diagram to show the separation of one speech channel from a hypergroup of 900 channels.

**A2** (a) Sketch (a) shows the essential stages of a group modulator. Each basic group signal with a frequency range of 60.6 to 107.7 kHz modulates a different carrier. The basic group signals are attenuated to the required level by attenuators before application to the balanced modulators, the attenuators also improve matching between the modulators and preceding stages. The output of each modulator stage is a double-sideband suppressed-carrier signal, of the form  $f_c \pm (60.6 \text{ to } 107.7 \text{ kHz})$ . Sideband filters select the lower sidebands,  $f_c - (60.6 \text{ to } 107.7 \text{ kHz})$ , which are then combined to form the supergroup. Sketch (b) shows the frequency spectrum of the assembled supergroup,  $f_1$  to  $f_5$  representing the carriers used. By careful choice of the carrier frequencies used, the bandwidth occupied by the supergroup can be minimised whilst still maintaining the 900 Hz guardband requirement.



(b) The carrier frequencies used in CCITT systems are 420, 468, 516, 564 and 612 kHz.

(c) Sketch (c) shows the principle of selecting one channel from a 900 channel hypergroup.



(c)

**Q3** A document is to be transmitted by an analogue facsimile system.

- describe THREE factors necessary to ensure synchronisation of transmitter and receiver
- explain the effect of variation of EACH factor in (a) on the correct reception of the document
- state the standard frequencies used in facsimile transmission over
  - audio cables
  - coaxial cables.

**A3 (a) Phasing** The scanning devices at the transmitter and receiver must occupy the same relative positions with regard to the transmitted picture and the receiving blank. Phasing pulses are used to align the scanners prior to picture transmission.

**Speed Synchronisation** The drum speeds of the two machines must be the same to maintain the same phase relationship between the scanning devices throughout picture transmission.

**Factor of Co-operation** This factor specifies the height-to-width ratio of the picture space of a machine; it depends on the drum diameter and the scanning intensity. To maintain the same height-to-width ratio between transmitted picture and received copy, the factors of co-operation of both machines must be the same.

**(b) Phasing** Misalignment of the transmit and receive scanners prior to picture transmission can result in the transmitted picture not projecting completely onto the recording blank. A split picture is a common result of a phasing error.

**Speed Synchronisation** A constant speed error between the two machines causes a skewing effect on the received copy. The direction and magnitude of the skew depends on the relative magnitude of the speed error.

**Factor of Co-operation** Different factors of co-operation cause the received copy to have an incorrect height-to-width ratio giving a distorted picture. The type and amount of distortion produced depends on the relative difference in the factors of the two machines.

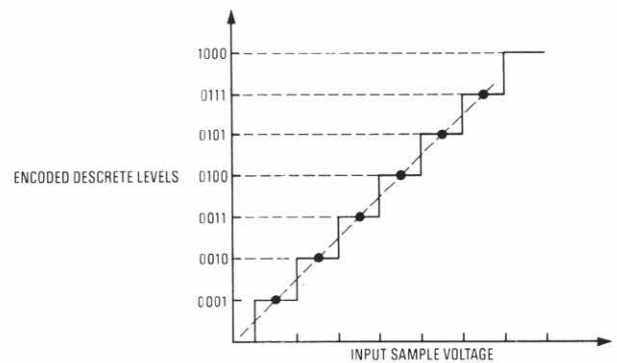
- $1300 \pm 550$  Hz
  - $1900 \pm 550$  Hz.

**Q4** For a pulse-code modulation (PCM) system

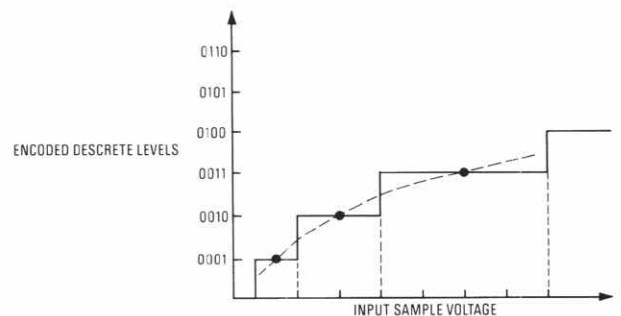
- Explain the term quantisation.
- Distinguish with the aid of sketches between linear and non-linear quantisation.
- Explain the term quantisation noise.
- By the use of sketches show the effect of linear and non-linear quantisation on the levels of quantisation noise.

**A4 (a)** Quantisation is a process where a continuous set of values is divided into a finite number of discrete levels. In PCM, all sample voltage amplitudes within a specified range (quantisation step) are encoded into a digital code (discrete value).

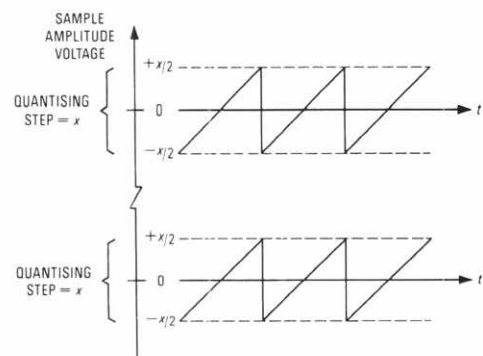
**(b)** In linear quantisation, the differences between the upper and lower limits of all quantisation steps are equal. Sketch (a) shows the transfer characteristics of an encoder using linear quantisation, where all quantisation steps are equal. In non-linear quantisation the quantising steps are unequal. Sketch (b) shows the transfer characteristics of an encoder using non-linear quantisation, the steps doubling in spread as the sample amplitudes increase.



(a)



(b)

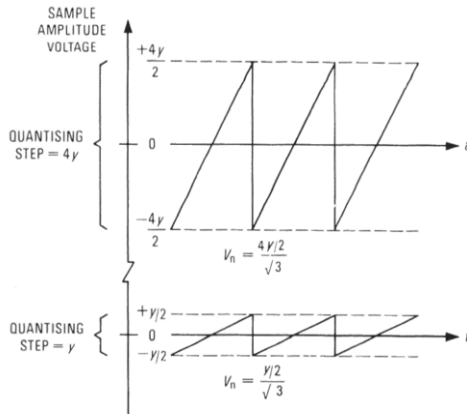


$$\text{RMS NOISE VOLTAGE ALL LEVELS } V_n = \frac{x/2}{\sqrt{3}}$$

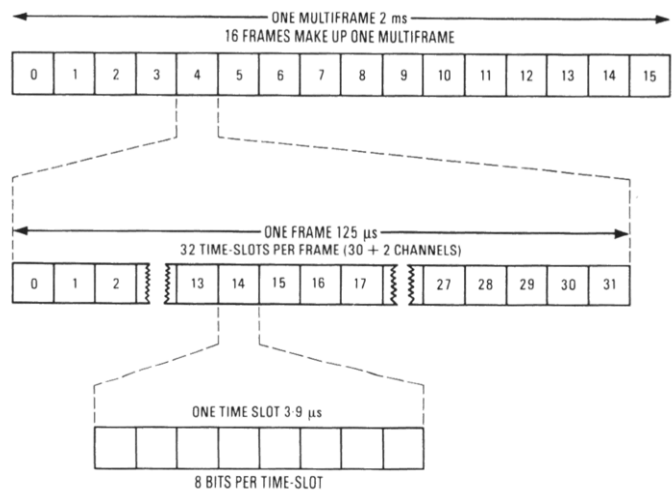
(c)

(c) Any sample whose amplitude lies within the upper and lower limits of a particular quantising level is encoded by using a binary code which corresponds to that level. The difference between the actual amplitude of the signal sample and the quantised value, which is generally taken as being midway between the upper and lower limits, is termed *quantisation noise*. Quantisation noise is a form of distortion introduced by the quantisation process.

(d) Sketches (c) and (d) show the triangular nature of quantisation noise produced by linear and non-linear quantisation. Non-linear quantisation gives a more constant signal-to-quantisation-noise ratio over the range of operation.



(d)



(b)

Note:

Time-slot 0 carries frame alignment pattern (even frames).

Time-slot 0 carries NOT word (odd frames).

Time-slot 16 carries signalling information except for frame 0 where multiframe alignment pattern is inserted.

Time-slots 1 to 15 and 17 to 31 carry telephone channel PCM signals.

frame, a NOT word is transmitted. Synchronisation is established when two frame alignment words and one NOT word are correctly received, during time-slot 0, over three consecutive frames. Multiframe alignment is achieved by transmission of a multiframe alignment pattern (word) in time-slot 16 of frame 0.

(c) Traditionally, PCM has been considered to be more economic over routes up to approximately 40 km. Where existing cable capacity is nearing exhaustion and heavy capital expenditure on new cables is required, PCM would appear to be an economic proposition. In general, PCM is more economic where bandwidth is not expensive and where equivalent performance cannot be obtained by the use of FDM.

**Q5** For a pulse-code modulated (PCM) line signal regenerator

(a) State the need for regenerators.

(b) Draw and fully label a block diagram.

(c) Sketch FIVE voltage waveforms to show how the incoming signal is regenerated.

**A5** See CGLI: Transmission T5 Option (1986), Supplement, April 1987, Q5, p. 11.

**Q6** For a CEPT 30 channel PCM system

(a) explain the effect of loss of synchronisation

(b) (i) draw a diagram to illustrate the frame and multiframe structure

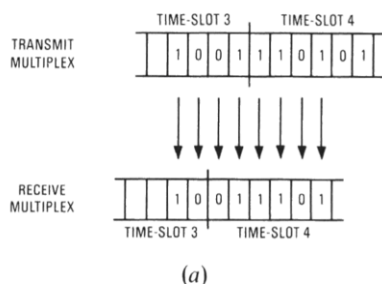
(ii) explain how synchronisation of the system is achieved

(c) state the conditions under which PCM transmission is more economic than frequency division (FDM) transmission.

**A6** (a) Loss of synchronisation causes the transmit and receive multiplexes to become out of step in time. The consequent misalignment of time-slots causes bits from the incoming bit stream to be routed into the wrong time-slot. Sketch (a) shows the result of loss of synchronisation.

(b) (i) Sketch (b) shows the frame and multiframe structure.

(ii) Synchronisation is maintained by the use of frame and multiframe alignment signals transmitted during specified time-slots. A frame-alignment pattern (word) is transmitted in time-slot 0 of every alternate frame. During time-slot 0 of the intervening



(a)

**Q7** (a) Explain, with the aid of sketches, why it is convenient to represent a uniform transmission line by T or  $\pi$  networks.

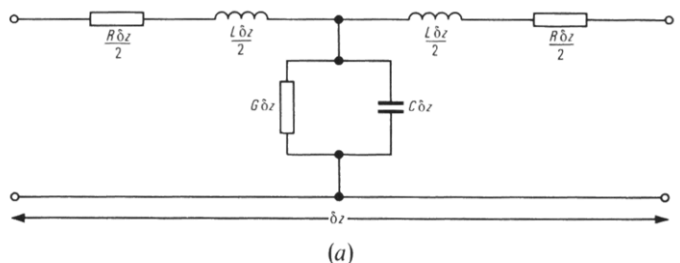
(b) Explain what is meant by the term characteristic impedance,  $Z_0$  of uniform transmission line.

(c) Calculate  $Z_0$  given that the open circuit impedance = 800  $\Omega$  and the short circuit impedance = 600  $\Omega$ .

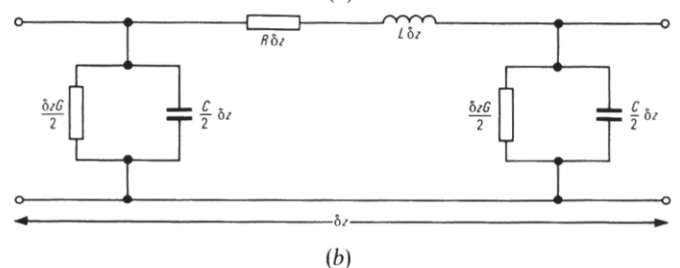
(d) State a formula for the magnitude of the current at any point on the transmission line in terms of the attenuation coefficient  $\alpha$  and the length of the line  $l$ .

(e) Given that  $\alpha = 2$  dB/km and the sending end current is 10 mA, plot a graph to show the relationship in (d) over the first 5 km of line.

**A7** (a) Sketches (a) and (b) show how a short length of uniform



(a)



(b)

transmission line can be represented as equivalent  $T$  and  $\pi$  networks in terms of its four primary line coefficients. Hence ordinary network theory can be used to develop equations which give the secondary coefficients of the line in terms of its primary coefficients.

(b) The characteristic impedance ( $Z_0$ ) of a uniform transmission line is the input impedance of an infinitely long length of that line.  $Z_0$  depends on the primary coefficients of the line and is in general a function of frequency.

(c) The characteristic impedance,  $Z_0$ , is given by

$$Z_0 = \sqrt{(Z_{oc} \times Z_{sc})},$$

where  $Z_{oc}$  is the open-circuit impedance, and  $Z_{sc}$  is the short-circuit impedance.

$$\therefore Z_0 = \sqrt{(800 \times 600)} = 692.8 \Omega.$$

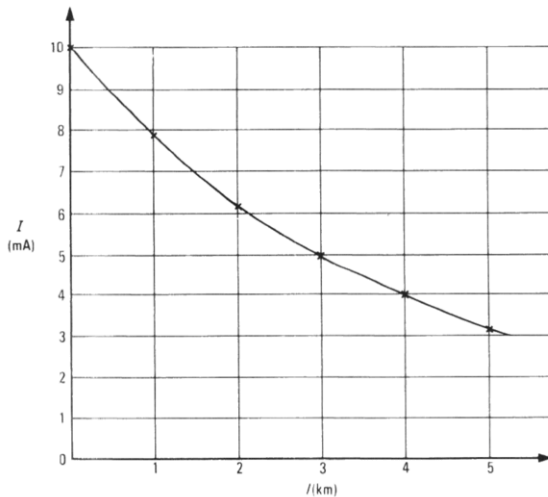
$$(d) \quad -\alpha l = 20 \log_{10} \left( \frac{I_l}{I_s} \right),$$

where  $I_l$  is the magnitude of the current  $l$  km from the input,  $I_s$  is the magnitude of the input current, and  $\alpha$  is the attenuation in dB/km.

$$\therefore I_l = I_s \text{ antilog}_{10} \left( \frac{-\alpha l}{20} \right).$$

(e) By using the equation in part (d), the magnitude of the current at 1 km spacings is as shown in the following table (see sketch (c)):

$l$ (km)	$I_l$ (mA)
0	10.0
1	7.9
2	6.3
3	5.0
4	4.0
5	3.2



(c)

**Q8** An audio cable pair has the following primary coefficients

$$R = 30 \Omega/\text{km} \quad C = 0.08 \mu\text{F}/\text{km}$$

$$L = 1 \text{ mH}/\text{km} \quad G \text{ can be neglected.}$$

The pair is loaded with 88 mH coils at 2 km intervals.

(a) Calculate the attenuation per km of the loaded pair at an angular velocity of  $\omega = 5000$  rad/s.

(b) Calculate the theoretical cut-off frequency.

(c) Estimate the upper usable frequency.

**A8** (a) With 88 mH coils spaced at 2 km intervals the effective line inductance becomes  $1 + 44$  mH per km.

The effective propagation coefficient,

$$\gamma' = \sqrt{\{R + j\omega L\}(G + j\omega C)}, \quad \text{where } L = 45 \text{ mH.}$$

$$(R + j\omega L) = 30 + j(5000 \times 45 \times 10^{-3}),$$

$$= 30 + j225,$$

$$= \sqrt{(30^2 + 225^2)} \angle \tan^{-1} \left( \frac{225}{30} \right),$$

$$= 227 \angle 82.4^\circ.$$

$$(G + j\omega C) = 0 + j(5000 \times 0.08 \times 10^{-6}),$$

$$= j(4 \times 10^{-4}),$$

$$= 4 \times 10^{-4} \angle 90^\circ.$$

$$\gamma' = \sqrt{\{227 \angle 82.4^\circ\} \times \{4 \times 10^{-4} \angle 90^\circ\}},$$

$$= \sqrt{(0.0908 \angle 172.4^\circ)} = 0.3013 \angle 86.4^\circ.$$

The effective attenuation coefficient  $\alpha'$  = the real part of  $\gamma'$ .

$$\therefore \alpha' = 0.3013 \cos 86.4^\circ = 0.02 \text{ Np/km.}$$

(b) Cut-off frequency,

$$f_c = \frac{1}{d\pi\sqrt{LC}},$$

where  $d$  is the distance between loading coils and  $L$  is the effective inductance.

$$\therefore f_c = \frac{1}{2\pi\sqrt{(45 \times 10^{-3} \times 0.08 \times 10^{-6})}},$$

$$= 2.65 \text{ kHz.}$$

(c) The upper usable frequency is approximately  $0.7f_c = 1.8 \text{ kHz.}$

**Q9** (a) Explain the meaning of the term optical waveguide.

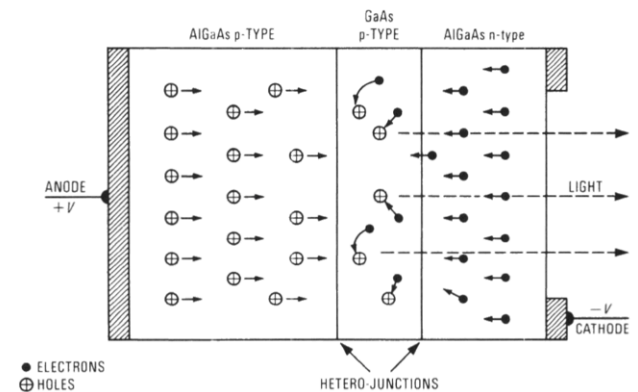
(b) Explain with the aid of sketches the construction and principle of operation of a light-emitting diode.

(c) Sketch the electroluminescent spectrum of a GaAs LED. Label both axes.

**A9** (a) An optical waveguide is a dielectric media used for the transmission of electromagnetic waves whose frequency lies in the light region of the electromagnetic spectrum.

(b) Sketch (a) shows the constructional features of a double heterojunction light-emitting diode (LED). The diode consists of p-type and n-type AlGaAs layers separated by a p-type GaAs layer. The n-type layer forms the cathode of the diode.

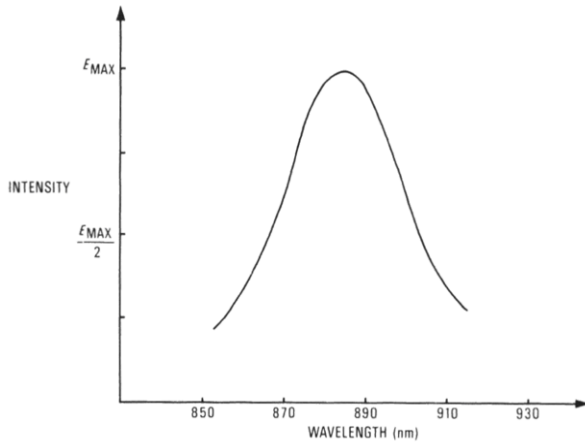
When the diode is operated in the forward bias condition, electrons in n-type layer cross the p-n junction and diffuse into the GaAs layer, where they become minority carriers. The injected electrons combine with the majority carriers (holes), the recombination producing photons of light. The photons are produced with an energy which corresponds to the bandgap energy



(a)

of the GaAs layer (energy =  $hf$ , where  $h$  is Planck's constant and  $f$  is the frequency of the emitted light). The p-p heterojunction barrier potential deters the electrons from entering the p-type AlGaAs layer, and so photon production is limited to the GaAs layer. Reabsorption of the emitted light is minimised because the bandgap energy of AlGaAs is much larger than that of GaAs.

(c) Sketch (b) shows the electroluminescent characteristics of a typical GaAs LED.



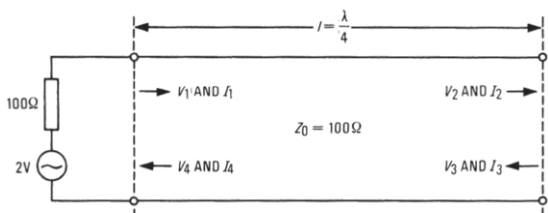
(b)

**Q10** At a particular frequency, a cable pair has a characteristic impedance of  $100 \Omega$  and is  $\lambda/4$  long.

When correctly terminated the line has a loss of 3 dB. A 2 V 100  $\Omega$  generator is connected to a line terminated in an open circuit

- determine the sending-end impedance
- draw phasor diagrams to scale to illustrate the incident and reflected current and voltages
  - at the sending end
  - at the termination.

**A10** (a) Refer to sketch (a). The sending-end impedance  $Z_s = V_s/I_s$ , where  $V_s$  and  $I_s$  are the vector sums of the incident and reflected voltages and currents respectively. As the source



(a)

impedance and the line  $Z_0$  are equal, the sending-end incident voltage and current are:

$$V_1 = \frac{E}{2} = 1 \angle 0^\circ \text{ V.}$$

$$I_1 = \frac{V_1}{Z_0} = 0.01 \angle 0^\circ \text{ A.}$$

In propagating to the open-circuit termination, the incident travelling waves undergo attenuation and phase shift. 3 dB loss in terms of voltage and current attenuation is

$$\text{antilog}_{10} \left( -\frac{3}{20} \right) = 0.707.$$

The phase shift is

$$-\beta l = -\frac{2\pi}{\lambda} \times \frac{\lambda}{4} = -\frac{\pi}{2} \text{ radians} = -90^\circ.$$

The incident voltage ( $V_2$ ) and current ( $I_2$ ) at the open circuit end are therefore:

$$V_2 = 0.707|V_1| \angle -90^\circ = 0.707 \angle -90^\circ \text{ V.}$$

$$I_2 = 0.707|I_1| \angle -90^\circ = 0.00707 \angle -90^\circ \text{ A.}$$

At the open circuit, the voltage wave is reflected in phase whilst the current wave is reflected in antiphase. The reflected voltage and current at the open-circuit end are therefore:

$$V_3 = 0.707 \angle -90^\circ \text{ V.}$$

$$I_3 = 0.00707 \angle +90^\circ \text{ A.}$$

In propagating back to the sending end, the waves undergo attenuation and phase change as before. The reflected voltage and current at the sending end are therefore:

$$V_4 = 0.707|V_3| \angle -180^\circ = 0.707^2 \angle -180^\circ = 0.5 \angle -180^\circ \text{ V.}$$

$$I_4 = 0.707|I_3| \angle 0^\circ = 0.005 \angle 0^\circ \text{ A.}$$

$V_1$  and  $V_4$  are in antiphase; hence

$$V_s = |V_1| - |V_4| = 1.0 - 0.5 = 0.5 \text{ V.}$$

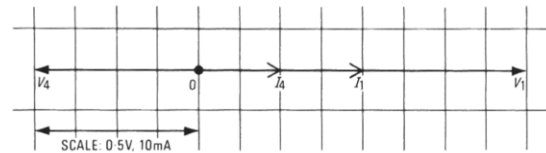
$I_1$  and  $I_4$  are in phase; hence

$$I_s = |I_1| + |I_4| = 0.01 + 0.005 = 0.015 \text{ A.}$$

Sending-end impedance

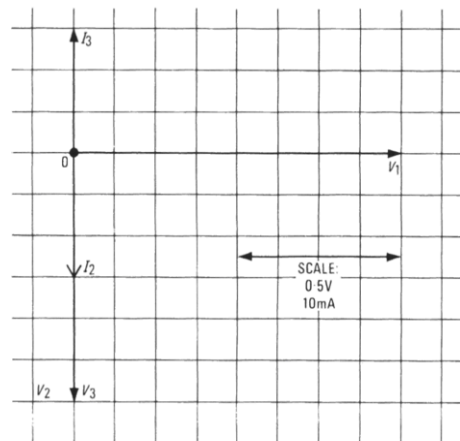
$$Z_s = \frac{V_s}{I_s} = \frac{0.5 \text{ V}}{0.015 \text{ A}} = 33.33 \Omega.$$

(b) Using the information in part (a), the phasor diagrams can be constructed, and are shown in sketches (b) and (c).



$V_1$  and  $I_1$  are the incident voltage and current.  
 $V_4$  and  $I_4$  are the reflected voltage and current.

(b) Sending end



$V_2$  and  $I_2$  are the incident voltage and current.  
 $V_3$  and  $I_3$  are the reflected voltage and current.

(c) Termination

# CGLI: MICROELECTRONIC SYSTEMS T4 OPTION (1986)

Students were required to answer any six questions. The time allowed was three hours. Students are advised to read the notes on p. 23

Some questions require reference to Tables 1, 2 and 3, which detail microprocessor instructions for different microprocessors, and which were attached to the examination paper. Because of limitations of space, these have not been included here; and, just to indicate the method, answers for only one microprocessor, the 6502 are given.

**Q1** Refer to Fig. 1 which shows a fundamental microprocessor system.

- Name the devices labelled A, B, C and D.
- Explain the function of block B.
- State FOUR on-chip facilities provided by block A.
- At a user interface, it is required to provide a  $4 \times 4$  matrix hexadecimal keyboard and a BCD 7 segment LED display. Describe in detail the hardware required to implement BOTH these devices.

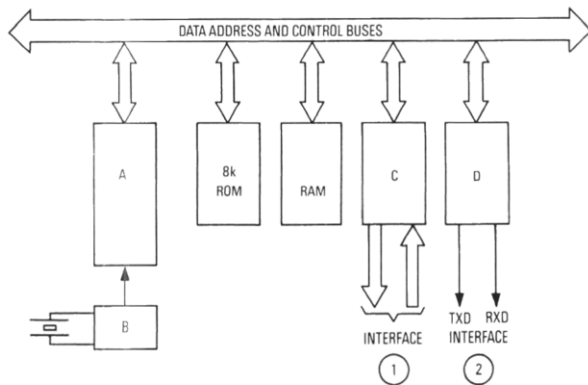


Fig. 1

- A1** (a) A: Microprocessor  
B: Clock oscillator  
C: Parallel interface (PIO, PIA or VIA)  
D: Serial interface (UART/USART)

(b) Block B is a timing generator for the microprocessor system. This generator is a crystal-controlled oscillator which provides a stable timing clock for the microprocessor and other system elements.

(c) The facilities provided by block A are:

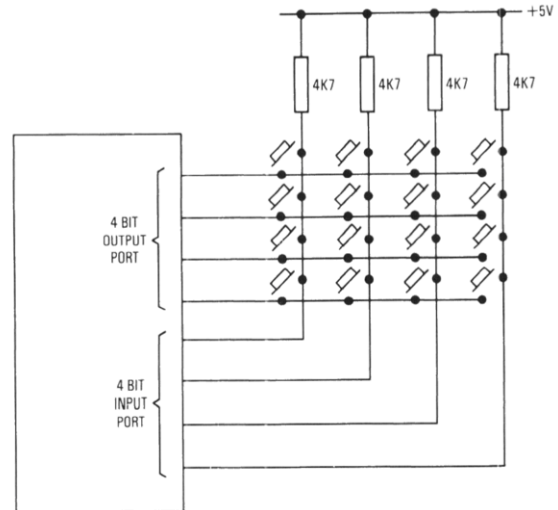
- accumulator/general-purpose registers,
- flag/status/condition code register,
- index register,
- stack pointer,
- program counter,
- arithmetic and logic unit,
- instruction decoder,
- clock generator,
- timing control,
- instruction decoder, and
- address/data bus buffers.

(d) A diagram of the user interface is shown in sketch (a).

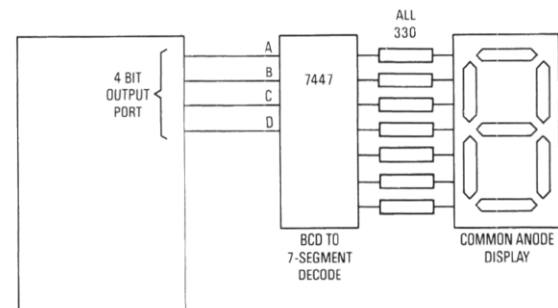
The matrix keypad columns are connected via pull-up resistors to the +5 V supply so that the 4 bit input port reads each bit as a 1. The 4 bit output port generates a 'walking zero' code so that any logic 0 received by the 4 bit input port indicates the column of a key closure. Since the row in which the walking zero is present is known at any instant, the precise key closed can be determined.

A diagram of the display interface is shown in sketch (b).

The display interface requires a seven-segment code to be sent to the display device. The microcomputer outputs a 4 bit binary coded decimal (BCD) code, and so an interface is required to convert this to the necessary seven-segment code (common anode in this case). Commercial devices are available to perform this function (for example 7447), and only a current limiting resistor is required for each light-emitting diode (LED) segment.



(a)



(b)

**Q2** Refer to the system in Fig. 1

(a) For a common microprocessor, draw the address decoding circuitry necessary to achieve the 'memory mapping' of the ROM within the address range E000H to FFFFH.

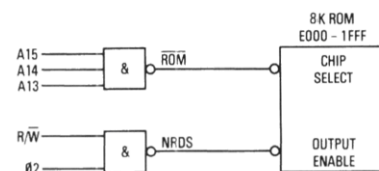
(b) Describe a hardware interface necessary for the system to drive a serial I/O device, without handshake.

**A2** Now,

	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
E000H:	1	1	1	0	0	0	0	0
FFFFH:	1	1	1	1	1	1	1	1
	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
E000H:	0	0	0	0	0	0	0	0
FFFFH:	1	1	1	1	1	1	1	1

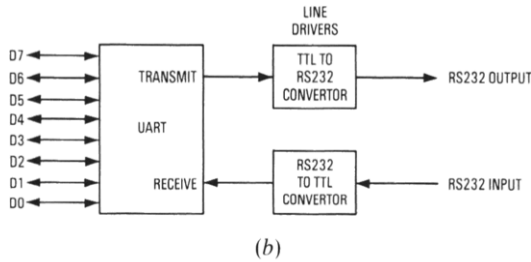
The three most significant bits (A<sub>15</sub>, A<sub>14</sub> and A<sub>13</sub>) must be used to provide the necessary decoding signal.

Thus for the 6502 microprocessor, the address decoding circuitry is shown in sketch (a).



(a)

(b) The microprocessor operates on parallel data, at 0 V and 5 V. Serial input/output devices usually operate on RS232 data which is at +12 V and -12 V. Therefore, the interface is required to convert between parallel and serial data, and to convert between transistor-transistor logic (TTL) and RS232. The former could be achieved by using a UART and the latter by suitable line drivers, sketch (b).



**Q3** If a microprocessor-based system were to be used as the basis of a 'real time controller' of several I/O devices, explain how each of the following could be implemented:

- (a) timing
- (b) servicing of the I/O devices.

Your explanations should include a statement of any additional hardware and software that may be required.

**A3 (a) Timing for a Real-Time Controller**

This could be implemented by using a real-time clock, which may be a dedicated device or derived from the master system clock.

A real-time clock could be implemented by using an interval timer. These devices usually contain timer registers which can be loaded and decremented under program control. A 'timeout' either interrupts the central processing unit (CPU) or conditions a flag which can be polled by the CPU.

Alternatively, discrete components can be used to divide down the master clock frequency to produce timed interrupts.

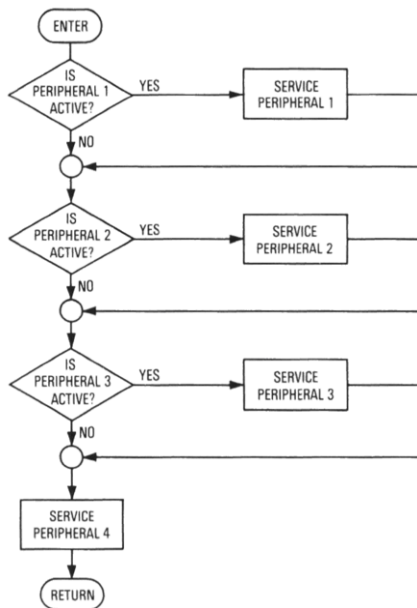
**(b) Servicing of Input/Output (I/O) Devices**

The I/O devices may be serviced by software polling or by interrupt I/O.

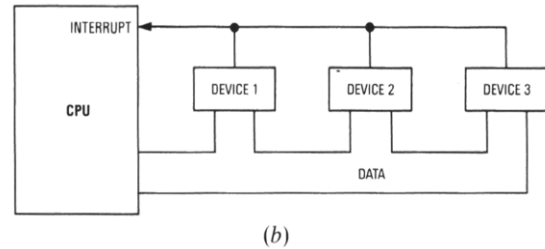
Software polling requires that the CPU examines its peripherals continuously, to determine whether any need servicing, see sketch (a).

This method is very wasteful of CPU time since most peripherals operate at a considerably slower speed than the microprocessor.

In contrast, interrupt I/O is initiated by the interrupting device



sending a signal to the microprocessor, which causes the microprocessor to suspend the current program and to service the I/O devices, sketch (b). Frequently, a number of I/O devices are connected to a single interrupt line, and the microprocessor must poll the devices after the interrupt to discover which device caused the interrupt. This technique requires each peripheral to be equipped with a hardware latch to allow for more than one peripheral requiring service at the same time.



- Q4** (a) Explain what is meant by 'a nested interrupt system'.  
 (b) State **THREE** factors that influence interrupt response time.  
 (c) The measurement of elapsed time often plays an important role in microprocessor 'real time' applications. Compare the relative merits of **TWO** common methods of producing elapsed times (delays).

**Q5** (a) State the meaning of **EACH** of the following terms related to semiconductor memory devices:

- (i) volatility
- (ii) cycle time
- (iii) random access
- (iv) read mostly.

(b) Explain what is meant by a 'bus driver' and explain why such devices are required in microprocessor systems.

**A5 (a) (i) Volatility**

Some memory devices lose their stored data at power-down. Such devices are said to provide volatile storage. Generally random-access memory (RAM) is volatile, whereas read-only memory is non-volatile. It should be noted, however, that CMOS RAM is sometimes backed up by a battery so that its contents can be preserved for perhaps several years.

**(ii) Cycle Time**

This is the time taken to read, modify and write data to a given location within a memory device. The read cycle time is the time taken to complete a read of a memory device, whilst the write cycle time is the time required to perform a memory write operation.

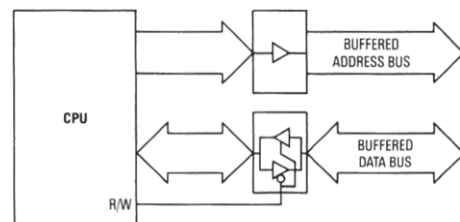
**(iii) Random Access**

Memory devices and systems which offer random access are characterised by all locations having identical access times, regardless of the physical location of data within a device.

**(iv) Read Mostly**

This term is often applied to EPROM/EAROM/EEPROM devices, which are only read devices in normal service. Occasionally though, these devices can be reprogrammed with fresh data.

(b) Typically, the output buffers of a microprocessor are only capable of driving a single transistor-transistor logic (TTL) load. When a number of devices are connected to a bus, an external buffer is required. A buffer should have high-input impedance, so that the bus is not significantly loaded, and a low-output impedance, so that a large drive current is obtainable. Such a device is known as a *bus driver*. These devices are usually tristate buffers, unidirectional for the address and control buses and bidirectional for the data bus (see sketch).



**Q6** (a) Distinguish between 'interrupts' and 'subroutines' as related to the execution of a program.

(b) A microprocessor system is to be used to control a domestic central heating system. Describe the requirements of such a system in terms of:

- (i) timing
- (ii) hardware
- (iii) software
- (iv) interfacing.

**A6** (a) When a subroutine is executed, the state of the program counter within the main program is saved on the stack. The program counter is then loaded with the start address of the subroutine. Normal fetch/execute is then resumed. At the end of the subroutine a RETURN instruction recalls the original program counter state and execution of the main program continues from the point at which it was left.

When an interrupt occurs, the instruction in progress at that instant is first completed, and then the program counter is loaded with the interrupt vectors, which specify the start address of the interrupt service subroutine, and fetch/execute is resumed. At the end of the interrupt service subroutine, a RETURN instruction recalls the original state of the program counter and execution of the main program is resumed.

Although the subroutine and interrupt responses are very similar, the subroutine is initiated at a point within the program defined by the user, whereas the interrupt can occur at any point during the execution of a program; the exact point within execution at which it occurs cannot be defined by the user.

**(b) (i) Timing**

Some form of real-time clock is required since the outputs of the system under control must change with time. A simple crystal-controlled oscillator would provide the basis of system timing.

**(ii) Hardware**

A single 8 bit processor would be more than adequate for this system. Only limited read-only memory (ROM) and very limited random-access memory (RAM) would be required and would allow address decoding components to be kept to a minimum. A single input/output (I/O) device comprising two data ports should provide enough I/O facilities for this system.

**(iii) Software**

It is unlikely that sophisticated control algorithms would be implemented. Simple on/off control is adequate for this application. A complex operating system is also unnecessary since a simple controller is most unlikely to require peripherals such as disc drives, printer, etc.

**(iv) Interfacing**

Fairly simple interfacing is required. Analogue-to-digital conversion is required for interfacing to temperature measurement transducers and the 'required temperature' potentiometer. Since the output ports would be incapable of providing sufficient power to operate solenoid valves, it is necessary to use power devices (transistors, thyristors, etc.) to switch the comparatively large currents involved.

**Q7** Fig. 2 shows a simplified diagram of a microprocessor with a parallel peripheral interface.

(a) List **FOUR** of the main components of a typical parallel peripheral interface.

(b) (i) Draw a flow chart for a program which produces, at one of the output port lines, a rectangular voltage waveform of 50 Hz frequency and mark space ratio 4:1. Assume a 4 ms delay subroutine already exists.

(ii) Use the flow chart in (b)(i) to write an assembly language program to carry out this task. Use one of the tables of instructions and codes—Table 1, Table 2 or Table 3—attached to the question

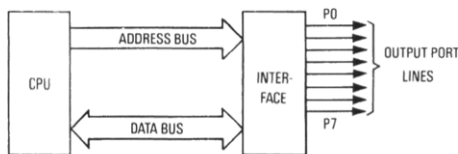


Fig. 2

paper. You should write out your program under the headings and using the format shown in Fig. 3.

PROGRAM TITLE: .....

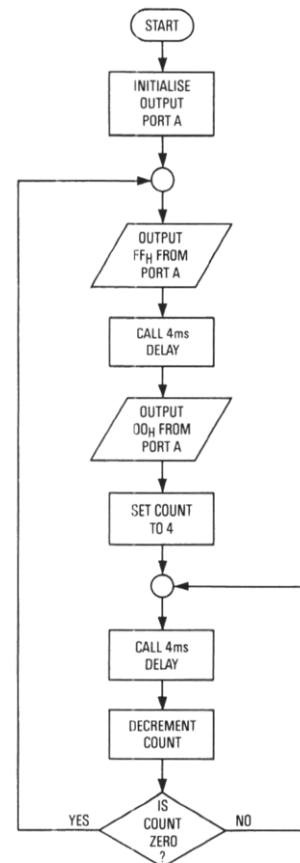
ADDRESS	DATA OR INSTRUCTION	LABEL	MNEMONIC	COMMENT

Fig. 3

**A7** (a) The main components of a typical parallel peripheral interface are:

- (i) data input register,
- (ii) data output register,
- (iii) data direction register,
- (iv) status register, and
- (v) control register.

(b) (i) **Flowchart**



**(ii) 6502 Assembly Language Program**

```

START LDA #FF
      STA DDRA ; Makes Port A all outputs
      STA PORTA ; Outputs a 1 (MARK) on all of Port A
LOOP JSR DELAY ; 4 ms delay
      INC PORTA ; Outputs a 0 (SPACE) on all of Port A
      LDX #04 ; Set count = 4
SPACE JSR DELAY ; 4 ms delay
      DEX ; count = count - 1
      BNE SPACE ; count = 0?
      DEC PORTA ; Outputs a 1 (MARK) on all of Port A
      JMP LOOP ; Next cycle
    
```

**Q8** (a) In relation to nested subroutines,

- state the importance of the stack
- describe how the stack is used.

(b) It is often required in certain applications, to display up-counts on numeric displays within microprocessor systems using time delays (loops).

(i) Draw a flow chart for a program which gives a time delay (loop) of a particular duration.

(ii) Use the flow chart in (b)(i) to write an assembly language subroutine for this delay. Use one of the tables of instructions and codes—Table 1, Table 2 or Table 3—attached to the question paper. You should write out your program under the headings and using the format shown in Fig. 3.

**A8** (a) **Nested Subroutines**

(i) **Importance of the Stack**

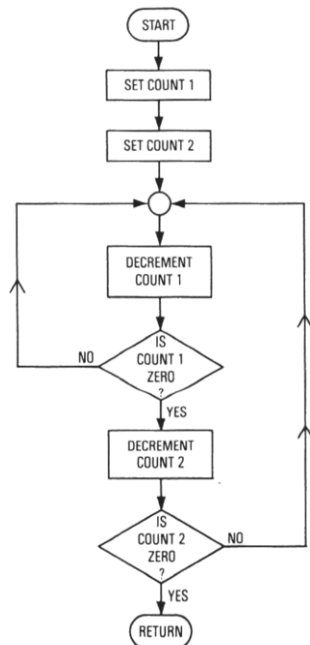
The stack is a 'last-in-first-out' (LIFO) register, which is often implemented as an area of memory. The stack can be used as a temporary storage area, particularly where the actual location of data is not as important as the order in which it was stored. The stack can be used to save many levels of return addresses for the support of subroutines and interrupts.

(ii) **Use of the Stack**

When a subroutine occurs, the return address within the main program must be saved on the stack. A RETURN instruction at the end of the subroutine restores the program counter from the stack, to allow the main program to continue. Now, if a subroutine itself calls another subroutine (nested subroutines) then the return address within the first subroutine is also saved on the stack. At the end of the second subroutine, the program counter is restored to the return address within the first. At the end of the first subroutine, the program counter state within the main program is restored.

A stack pointer register within the microprocessor holds the address of the current top of the stack.

(b) (i) **Program Flowchart**



(ii) **Assembly Language Program**

```

START LDX #VAL1 ; Set count 1
      LDY #VAL2 ; Set count 2
LOOP  DEX      ; count 1 = count 1 - 01
      BNE LOOP ; Loop until count 1 is zero
      DEY      ; count 2 = count 2 - 01
      BNE LOOP ; Loop until count 2 is zero
      RTS      ; Return from subroutine
    
```

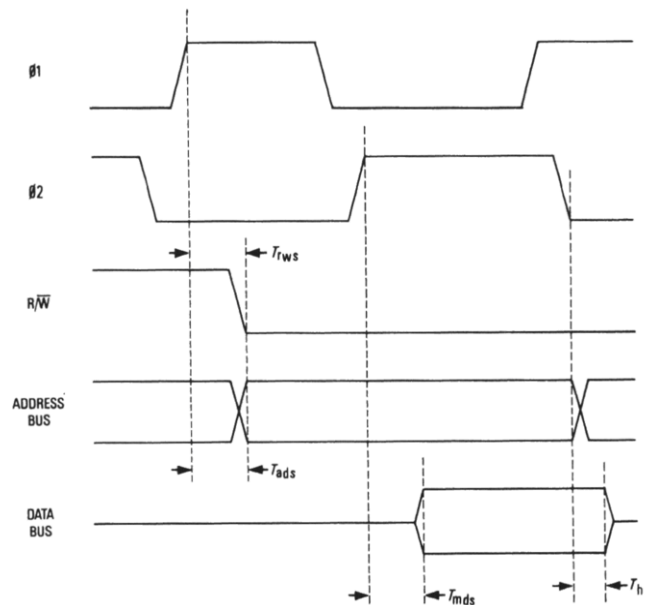
Clearly the actual delay generated will depend upon the values VAL1 and VAL2 used as well as the clock frequency of the system.

**Q9** With the aid of timing diagrams, explain EACH of the following:

- a memory write instruction
- a non-maskable interrupt.

**A9** (a) **6502 Memory Write**

Sketch (a) shows the write timing diagram for a 6502.



(a)

The read/write ( $R/\bar{W}$ ) control line begins to go LOW on a rising edge of  $\phi_1$ , the transition becoming stable after a short period, called the *read/write set up time* ( $T_{rws}$ ). Similarly, address changes are initiated by a rising edge of  $\phi_1$  and the address bus becomes stable after a similar set up time (*address set up time*,  $T_{ads}$ ). The next rising edge of  $\phi_2$  begins the transfer of data from the central processing unit to memory. Stable data is present on the data bus after another short delay, the *write data set up time* ( $T_{mds}$ ). The data is latched into the memory device on the next falling edge of  $\phi_2$ , remaining stable for a short hold time following this edge. This is called the *write data hold time* ( $T_h$ ).

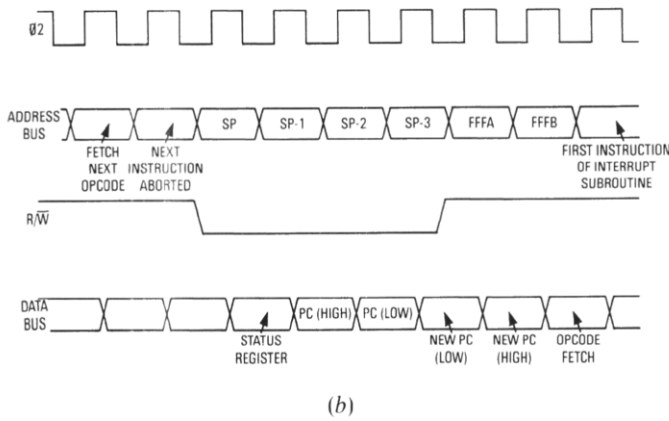
Typical values, for a 1 MHz clock, are shown below:

Parameter	Minimum (ns)	Typical (ns)	Maximum (ns)
$T_{rws}$	—	100	300
$T_{ads}$	—	200	300
$T_{mds}$	—	150	200
$T_h$	10	30	—

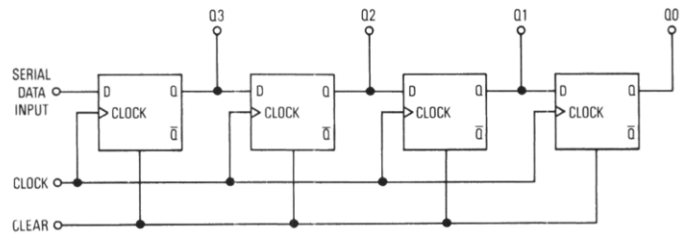
(b) **6502 Non-Maskable Interrupt**

Some interrupt inputs can, under certain circumstances, be ignored by the microprocessor. Those which cannot be ignored are called *non-maskable interrupts* (NMIs). The 6502 NMI input is edge triggered by a negative-going transition. Clearly then this input must be returned to a logic 1 before any further interrupts can be accepted.

The CPU completes the instruction in progress at the time that the interrupt occurred. The program counter and status register are pushed onto the stack and the contents of locations FFFAH and FFFBH are loaded into the program counter. This is the start address of the interrupt service subroutine. At the end of the interrupt service subroutine, a RETURN FROM INTERRUPT instruction will PULL the status register and program counter from the stack and the interrupted program continues. (See sketch (b).)



So, to transmit data from the microprocessor to a serial peripheral, it is necessary to load the shift register in parallel and to shift data out in serial form. Conversely, for the microprocessor to read serial data from the peripheral the shift register must shift in the data in serial form to allow the microprocessor to read a parallel dataword. (See sketch (a).)



**Q10** (a) Explain the meaning of each of the following terms:

- (i) real-time processing
- (ii) mnemonic code
- (iii) fetch cycle
- (iv) push operation.

(b) Explain how a shift register can be used to communicate with a peripheral which accepts serial data.

(c) Describe ONE common format used for sending data from a microcomputer system to a serial peripheral device.

**A10** (a)

**(i) Real-Time Processing**

This refers to where any data is processed as soon as it becomes available and results become available within a specific time limit. For example, a microprocessor-based car anti-lock braking system would require real-time processing, since there must be minimal delay between receiving data and altering system outputs.

**(ii) Mnemonic Code**

Although the microprocessor can only recognise groups of binary numbers, it is unusual for a programmer to write directly in binary since this is very prone to error. Usually a hexadecimal equivalent of the binary is used, but even this is not particularly readable. It is usual, therefore, initially to write programs in assembly language. Each assembly-language statement comprises a two-, three- or four-letter mnemonic code which represents the operation code (op-code); for example ADD for 'add to the accumulator', accompanied by an operand which may be zero, one or two further bytes.

**(iii) Fetch Cycle**

The operation of the microprocessor is cyclic. Every cycle begins by fetching a data word from memory. This data word is then decoded by the microprocessor and the appropriate action taken (for example, load the accumulator from a memory location). Since the microprocessor is concerned only with internal operations during an instruction decode period, manufacturers often provide some means of detecting these periods to allow buses to be used for other activities (for example, refreshing dynamic random-access memory).

**(iv) Push Operation**

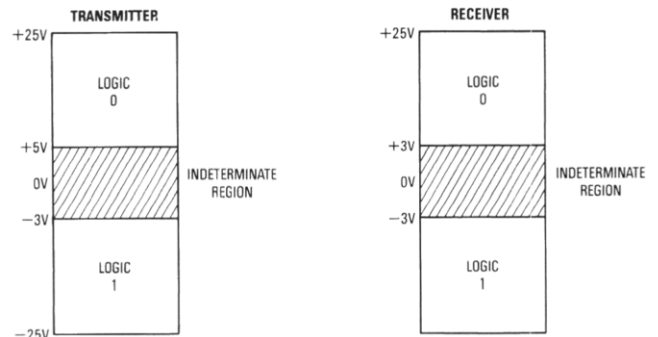
A 'push' is an instruction which causes the contents of a register to be duplicated on the top of the stack. This instruction can be used to save temporary data, etc. The microprocessor will itself push return addresses when a subroutine or interrupt occurs to allow the main program to be resumed from the precise point at which it was suspended.

(b) A shift register is made up from a number of flip-flops (bistables). Each flip-flop has an individual input and an individual output; and so they may be connected to provide one of the following structures:

- (i) parallel input, serial output,
- (ii) parallel input, parallel output,
- (iii) serial input, serial output, and
- (iv) serial input, parallel output.

Shift registers are often included on an interfacing device.

(c) Probably the commonest serial data format is the Electronics Industry Association (EIA) RS232C. This interface standard allows serial information to be transmitted in both directions simultaneously, although transmission is unidirectional along any individual line. There is also provision for synchronous or asynchronous communication and for handshaking if required. The signal levels are shown in sketch (b).



A logic 1 is represented by a voltage level lower than a logic 0. Special interfacing devices are available for the conversion of RS232C data to transistor-transistor logic (TTL) levels and vice versa.

The RS232C interface is designed for communication between the computer and a single peripheral, and so a number of such interfaces may be required.

The RS232C standard defines connections for a standard 25-way connector but it is unusual for the full RS232C to be implemented. A common subset is shown below:

Pin	Name	Function
1	Protective ground	Chassis ground
2	Transmit data	Serial data transmission line
3	Receive data	Serial data receive line
6	Data set ready	Indicates that the device is ready to receive data
7	Logical ground	Ground return for data signals
8	Data carrier detect	Indicates that a remote device is ready for data transmission

Answers contributed by P. A. Quinsey